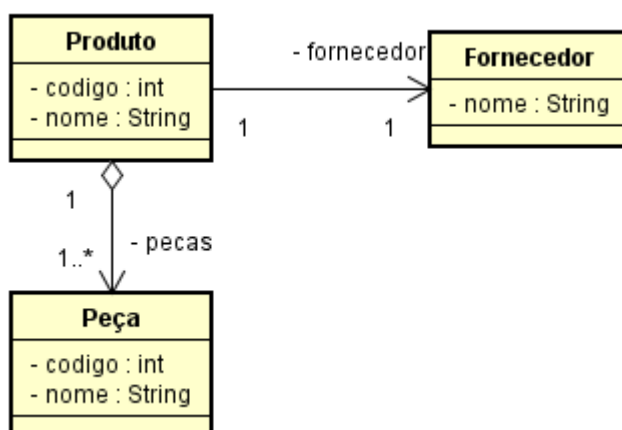


Aluno(a) : _____

1) Faça o que se pede(4,0):

- Modele uma classe **Fabricante** (representando no diagrama) com o atributo nome, indicando no diagrama que cada peça é fabricada por um único Fabricante. Modele também uma classe **Pessoa**, com cnpj e um método abstrato **mostrar** (sem parâmetro e retornando um String), como superclasse de Fornecedor e de Fabricante.
- Implemente as classes representadas no diagrama, inclusive Fabricante e Pessoa, com seus relacionamentos, atributos e métodos. Os construtores de cada uma devem atualizar todos os atributos da classe.
- Implemente um método na classe Produto para adicionar peças a um produto. O método recebe uma peça como parâmetro.
- Crie uma classe **Principal** que instancie um produto de um fornecedor com duas peças, cada uma com um fabricante diferente. Faça os vínculos necessários entre os objetos e as chamadas de métodos necessárias, conforme a representação do modelo após o item a).



- (2,0) Modele, através de um **diagrama de classes** o cenário de um carrinho de compras. Um carrinho de compras tem uma data, um código e as informações do cliente (nome e identidade). É formado por 1 ou mais itens de compra, sendo que cada item tem como característica o código, o valor e o nome. Olhando o item separadamente não é possível ver a que carrinho ele pertence mas olhando a partir do carrinho, é possível identificar todos os seus itens. As operações que podem ser realizadas em um carrinho de compra é: adicionar item e retirar item. Modele o diagrama com as classes candidatas, seus atributos, métodos, relacionamentos, navegabilidade e multiplicidade, conforme o cenário descrito.
- (2,0) Responda V ou F:
 - Testes é uma das atividades no processo de desenvolvimento de software. O desenvolvimento orientado a testes é uma metodologia onde, primeiro são elaborados os casos de testes e depois as funcionalidades do sistema, permitindo o refatoramento do código.
 - Exceção é um evento que interrompe o fluxo normal de um sistema. O java obriga o tratamento de exceções de algumas operações, e outras, não obrigatórias, podem ser tratadas para que o sistema não apresente o erro para o usuário. Elas são chamadas respectivamente de *unchecked* e *checked*.
 - Duas classes podem estar relacionadas através de um relacionamento todo-parte de agregação ou de composição. No primeiro, a parte é dependente do todo e no segundo não.

Aluno(a) : _____

- () Classes Abstratas e Interface não podem ser usadas para instanciar objetos mas podem ser usadas com o objetivo de reaproveitar código através de herança.
- () Uma *Thread* é uma linha de execução paralela mas para ela ser iniciada é necessário que outra Thread da mesma classe também seja executada simultaneamente.
- () *Junit*, *Jar* e *JavaDoc* são respectivamente um *framework* para testes, uma biblioteca de classes e ferramenta de documentação.
- () A criação de Classes Abstratas é uma forma de organizar os dados generalizando operações e características comuns de várias classes concretas.
- () *Design Pattern* é uma solução geral para um problema que ocorre com frequência dentro de um determinado contexto. São boas práticas formalizadas para projeto e implementação. Em GoF (*Gang of Four*), eles foram divididos nas categorias Inicial, Organizacional e Comportamental.
- () Os diagramas de casos de uso e classes são importantes na documentação de um sistema. O diagrama de classes é o principal diagramas para levantamento de requisitos com os usuários.
- () Utilizando herança podemos fazer com que um mesmo objeto se comporte de formas diferentes na aplicação. Esta ideia se refere aos conceitos de abstração e encapsulamento.

- 4) (2,0) Considerando que uma concessionária deseja controlar as revisões realizadas mantendo os dados referentes aos veículos, seus proprietários e o histórico das revisões de cada veículo, registrando quando foi realizada, quem fez a revisão e uma breve descrição. Para atender a esses requisitos, foi modelado o diagrama de classes abaixo. Implemente as classes, seus atributos e relacionamentos de acordo com a representação do diagrama. Crie construtores para a classe Cliente e Pessoa inicializando todos os seus dados, métodos de acesso (get e set) para a descrição da Revisão. Crie também um método para adicionar uma revisão a um veículo.

